

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение  
высшего образования  
«Северный (Арктический) федеральный университет имени М.В. Ломоносова»

Высшая школа информационных технологий и автоматизированных систем

## ЛАБОРАТОРНАЯ РАБОТА №3

По дисциплине: Защита информации в системах управления базами данных

На тему Кеширование Redis/memcached

Выполнил обучающийся:

Грозов Илья Владимирович

Направление подготовки / специальность:

10.03.01 Информационная безопасность

Курс: 3

Группа: 151113

Руководитель: Зубарев Александр Андреевич, ст.  
преподаватель

Отметка о зачете

Руководитель

А.А. Зубарев.

Архангельск 2024

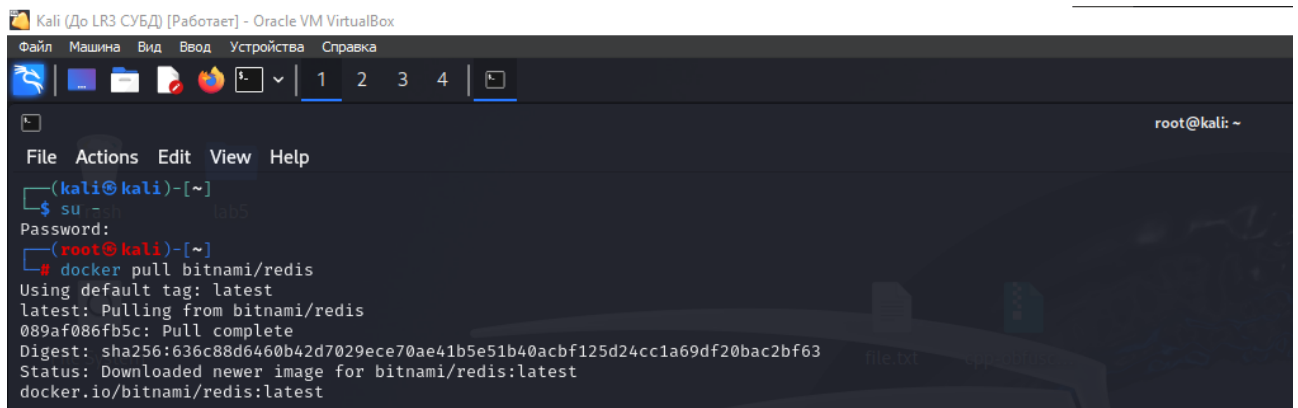
## ЗАДАНИЕ

Получить практический навык использования кеширования  
redis/memcached

## ХОД РАБОТЫ

### 1. КОНФИГУРАЦИЯ BITMANI

Для установки bitnami/redis на docker необходимо воспользоваться командой: `docker pull bitnami/redis`. Установка bitnami/redis на docker предоставлена на рисунке 1



```
Kali (До LR3 СУБД) [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
1  2  3  4
root@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ su -
Password:
(root@kali)-[~]
# docker pull bitnami/redis
Using default tag: latest
latest: Pulling from bitnami/redis
089af086fb5c: Pull complete
Digest: sha256:636c88d6460b42d7029ece70ae41b5e51b40acbf125d24cc1a69df20bac2bf63
Status: Downloaded newer image for bitnami/redis:latest
docker.io/bitnami/redis:latest
```

Рисунок 1 – Установка bitnami/redis на docker

По завершению установки bitnami/redis на docker необходимо запустить сервер redis. Запуск сервера redis отображен на рисунке 2



```
(root@kali: ~)
redis-server
2507:~ 12 Apr 2024 04:16:54.565 # 000000000000 Redis is starting 000000000000
2507:~ 12 Apr 2024 04:16:54.565 # Redis version=7.0.15, bits=64, commit=00000000, modified=0, pid=2507, just started
2507:~ 12 Apr 2024 04:16:54.565 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf
2507:~ 12 Apr 2024 04:16:54.565 # Increased maximum number of open files to 10032 (it was originally set to 1024).
2507:~ 12 Apr 2024 04:16:54.565 # monotonic clock: POSIX clock_gettime

Redis 7.0.15 (00000000/0) 64 bit
Running in standalone mode
Port: 6379
PID: 2507
https://redis.io

2507:~ 12 Apr 2024 04:16:54.569 # Server initialized
2507:~ 12 Apr 2024 04:16:54.569 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
2507:~ 12 Apr 2024 04:16:54.572 # Ready to accept connections
```

Рисунок 2 – Запуск сервера redis

Запустим отдельный контейнер под redis на docker при помощи команды: `docker run --name my-redis-container -it bitname/redis bash`. Запуск контейнера предоставлен на рисунке 3

```
(root@kali)-[~]
# docker run --name my-redis-container -it bitnami/redis bash
Using default tag: latest
latest: Pulling from bitnami/redis
redis 08:18:19.54 INFO =>
redis 08:18:19.55 INFO => Welcome to the Bitnami redis container
redis 08:18:19.55 INFO => Subscribe to project updates by watching https://github.com/bitnami/containers
redis 08:18:19.55 INFO => Submit issues and feature requests at https://github.com/bitnami/containers/issues
redis 08:18:19.56 INFO => Upgrade to Tanzu Application Catalog for production environments to access custom-c
n result reports, and VEX documents. To learn more, visit https://bitnami.com/enterprise
redis 08:18:19.56 INFO =>
```

Рисунок 3 – Запуск контейнера

### 1.1 Запись ключей в redis

Для записи ключей в redis необходимо произвести удаленное подключение в redis-cli. Запись ключей производится при помощи команды set <имя ключа>, <значение>. Подключение к redis и запись ключей отображено на рисунке 4

```
(root@kali)-[~]
# redis-cli

127.0.0.1:6379> set key1 value1
OK
127.0.0.1:6379> set key2 value2
OK
127.0.0.1:6379> █
```

Рисунок 4 – Подключение к redis и запись ключей

### 1.2 Извлечение ключей при помощи redis-cli

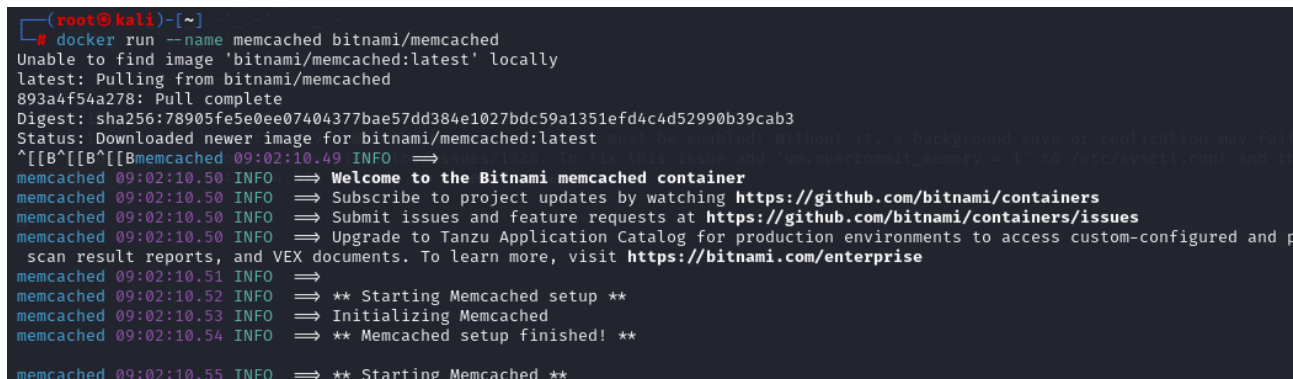
Для получения значений ключей из redis необходимо использовать команду: get <имя ключа>. Получение значений ключей отображено на рисунке 5

```
OK
127.0.0.1:6379> get key1
"value1"
127.0.0.1:6379> get key2
"value2"
127.0.0.1:6379> █
```

Рисунок 5 – Получение значений ключей

## 2. КОНФИГУРАЦИЯ MEMCACHED

Для установки memcached на docker необходимо воспользоваться командой: `docker run --name memcached bitnami/memcached/`. Установка memcached на docker отображена на рисунке 6



```
(root@kali)-[~]
# docker run --name memcached bitnami/memcached
Unable to find image 'bitnami/memcached:latest' locally
latest: Pulling from bitnami/memcached
893a4f54a278: Pull complete
Digest: sha256:78905fe5e0ee07404377bae57dd384e1027bdc59a1351efd4c4d52990b39cab3
Status: Downloaded newer image for bitnami/memcached:latest
^[[B^[[B^[[Bmemcached 09:02:10.49 INFO ==>
memcached 09:02:10.50 INFO ==> Welcome to the Bitnami memcached container
memcached 09:02:10.50 INFO ==> Subscribe to project updates by watching https://github.com/bitnami/containers
memcached 09:02:10.50 INFO ==> Submit issues and feature requests at https://github.com/bitnami/containers/issues
memcached 09:02:10.50 INFO ==> Upgrade to Tanzu Application Catalog for production environments to access custom-configured and p
scan result reports, and VEX documents. To learn more, visit https://bitnami.com/enterprise
memcached 09:02:10.51 INFO ==>
memcached 09:02:10.52 INFO ==> ** Starting Memcached setup **
memcached 09:02:10.53 INFO ==> Initializing Memcached
memcached 09:02:10.54 INFO ==> ** Memcached setup finished! **
memcached 09:02:10.55 INFO ==> ** Starting Memcached **
```

Рисунок 6 – Установка memcached на docker

Запуск контейнера memcached осуществляется при помощи команды: `docker run -d --name my-memcached-container memcached`. Запуск контейнера отображен на рисунке 7



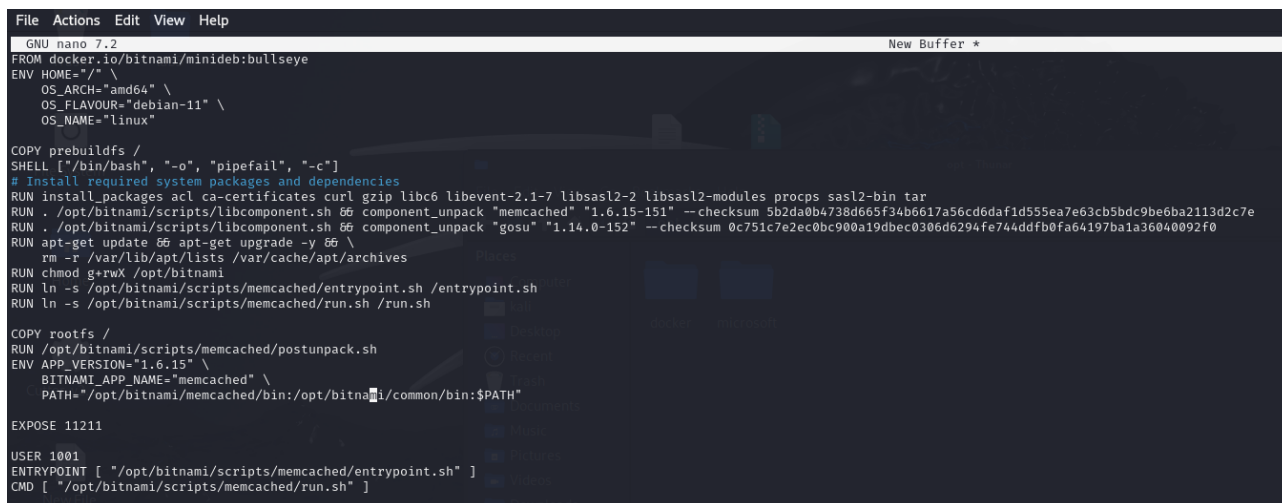
```
(root@kali)-[~]
# docker run -d --name my-memcached-container memcached
Unable to find image 'memcached:latest' locally
latest: Pulling from library/memcached
13808c22b207: Pull complete
131f56ef3fe0: Pull complete
aa35112bf68c: Pull complete
4a672c032d4a: Pull complete
e01135bb6895: Pull complete
f48f53906ef4: Pull complete
Digest: sha256:224efbcd38930137c6b877805174384ddbd896615c4c19f396d58e6b6287543d
Status: Downloaded newer image for memcached:latest
00e3bf3c4aab71475fadb0ec922c9c95e781a66e212443efc1063ed1346e6ef1
```

Рисунок 7 – Запуск контейнера

Сконфигурировать контейнер (собрать образ) также можно при помощи файла предложенного к работе. Для этого нужно в директории разместить файл с предложенным содержимым к работе. В терминале из директории прописать команду: `docker build -t gugenot/memcached:v1 .`

После сборки образа его можно запустить при помощи команды `docker run -d -p ip:port gugenot/memcached:v1`

Конфигурация для `docker` отображена на рисунке 8



```
File Actions Edit View Help
GNU nano 7.2
FROM docker.io/bitnami/minideb:bullseye
ENV HOME="/" \
    OS_ARCH="amd64" \
    OS_FLAVOUR="debian-11" \
    OS_NAME="linux"

COPY prebuildfs /
SHELL ["/bin/bash", "-o", "pipefail", "-c"]
# Install required system packages and dependencies
RUN install_packages acl ca-certificates curl gzip libc6 libevent-2.1-7 libsasl2-2 libsasl2-modules procps sasl2-bin tar
RUN . /opt/bitnami/scripts/libcomponent.sh 66 component_unpack "memcached" "1.6.15-151" --checksum 5b2da0b4738d665f34b6617a56cd6daf1d555ea7e63cb5bdc9be6ba2113d2c7e
RUN . /opt/bitnami/scripts/libcomponent.sh 66 component_unpack "gosu" "1.14.0-152" --checksum 0c751c7e2ec0bc900a19dbec0306d6294fe744ddfb0fa64197ba1a36040092f0
RUN apt-get update 66 apt-get upgrade -y 66 \
    rm -r /var/lib/apt/lists /var/cache/apt/archives
RUN chmod g+rwX /opt/bitnami
RUN ln -s /opt/bitnami/scripts/memcached/entrypoint.sh /entrypoint.sh
RUN ln -s /opt/bitnami/scripts/memcached/run.sh /run.sh

COPY rootfs /
RUN /opt/bitnami/scripts/memcached/postunpack.sh
ENV APP_VERSION="1.6.15" \
    BITNAMI_APP_NAME="memcached" \
    PATH="/opt/bitnami/memcached/bin:/opt/bitnami/common/bin:$PATH"

EXPOSE 11211

USER 1001
ENTRYPOINT [ "/opt/bitnami/scripts/memcached/entrypoint.sh" ]
CMD [ "/opt/bitnami/scripts/memcached/run.sh" ]
```

Рисунок 8 – Конфигурация

## 2.1 Запись ключей в `memcached` с TTL 5

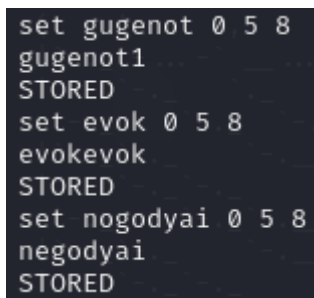
Подключимся к `memcached` по `telnet 127.0.0.1 11211`

Запишем в `memcached` следующие ключи с TTL5:

- `gugenot 0 5 8` со значением `gugenot1`;
- `evok 0 5 8` со значением `evokevok`;
- `negodyai 0 5 8` со значением `negodyai1`.

Где <имя ключа> флаг TTL длина значение.

Запись ключей предоставлена на рисунке 9



```
set gugenot 0 5 8
gugenot1
STORED
set evok 0 5 8
evokevok
STORED
set negodyai 0 5 8
negodyai
STORED
```

Рисунок 9 – Запись ключей в `Memcached`

## 2.2 Удаление записанных ключей

По истечению 5 секунд проверим при помощи команды `stats items`, что записанные ключи были удалены. Проверка на удаление ключей предоставлена на рисунке 10

```
stats items
END
```

Рисунок 10 – Проверка на удаление ключей

## 2.3 Запись в memcached ключа key5 со значением типа "int" равным числу 5

Запишем ключ при помощи команды: `set gugenot 0 30 1` значением 5. Запись ключа отображена на рисунке 11

```
set key5 0 30 1
5
STORED
```

Рисунок 11 – Запись ключа со значением 5

### 2.3.1 Увеличение значения ключа

В Memcached для увеличения значения ключа необходимо использовать команд `incr` – отвечающую за увеличение значения, `gugenot` – имя ключа, 5 – значение, на которое мы увеличиваем значение. Команда `incr gugenot 5` увеличит существующее значение 5 ключа `gugenot` на 5. Ключ `gugenot` со значением 10 представлен на рисунке 12

```
incr key5 5
10
```

Рисунок 12 – Ключ gugenot со значением 10

## КОНТРОЛЬНЫЕ ВОПРОСЫ

### **1. Приведите примеры проблем, которые может решить кэширование?**

Кэширование может существенно сократить время доступа к данным, уменьшая нагрузку на базу данных и сеть, данные могут быть временно сохранены в кэше – сервер базы данных будет менее загружен запросами на чтение

### **2. Приведите типичные уязвимости таких систем?**

Недостаточная проверка введенных пользователем данных, некорректное вычисление параметров буфера, незащищенные инсталляции



## ВЫВОД

В ходе выполнения лабораторной работы по теме: «Кеширование redis/memcached» получили практический навык использования кеширование redis/memcached